MULTIMEDIA UNIVERSITY

STUDENT ID NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 1, 2016/2017

TCP2201 OBJECT ORIENTED ANALYSIS AND DESIGN

(All sections/groups)

15 OCTOBER 2016
9:00 A.M – 11:00 A.M
(2 Hours)

INSTRUCTIONS TO STUDENTS

1. This question paper consists of seven (7) pages only including the cover page.
2. There are four (4) questions in this paper.
3. Answer **ALL QUESTIONS**.
4. All questions carry equal marks (25 marks) and the distribution of the marks for each sub-question is given.
5. Write your answers in the answer booklet provided.

Question 1

The following Java code for a Box class is given.

```
class Box {
      protected double width;
      protected double height;
      protected double depth;

      Box() {
      }
      Box(double w, double h, double d) {
            width = w;
            height = h;
            depth = d;
      }
      public void getVolume() {
            System.out.println("Volume = " + width *
                                        height * depth);
      }
}
```

a) Create a **SmallBox** class that inherits from the **Box** class above. The **SmallBox** class should have an additional private data member **weight**, a default constructor (which initializes all variables to zero), and a constructor with parameters. The main method and the expected output are also given below:

```
public static void main(String args[]) {
   SmallBox sm1 = new SmallBox (10, 7, 4, 33.3);
   System.out.println("width  = " + sm1.width);
   System.out.println("height = " + sm1.height);
   System.out.println("depth  = " + sm1.depth);
   System.out.println("weight = " + sm1.weight);
   sm1.getVolume();
}
```

```
width  = 10.0
height = 7.0
depth  = 4.0
weight = 33.3
Volume = 280.0
```

[8 marks]

b) Explain the concept of "*upcasting*". Give an example to illustrate your answer (you may use the objects from (a) or your own example objects).

[4 marks]

Continued...

c) In deciding whether to use an abstract class or an interface object, give reasons for choosing each of them. Show in your answer an example of how each is declared using Java.

[6 marks]

d) What is the output of the following program when compiled and executed?

```
class PC{
      int getMe(){ return 3; }
}
class CC extends PC{
      @Override
      int getMe(){ return 6; }
      int getMe(int var){ return var; }

      public static void main(String[] a){
            CC cc=new CC();
            System.out.println(cc.getMe());
            System.out.println(cc.getMe(5));
      }
}
```

[1 mark]

e) Briefly explain the object-oriented concepts of polymorphism being demonstrated in the code which gives the output shown in your answer for (e).

[4 marks]

f) Explain how `@Override` functions in the program above

[2 marks]

Continued...

Question 2

a) Briefly describe the work involved in the following types of software projects
   i. Corrective projects
   ii. Adaptive projects
   iii. Enhancement projects
   iv. Perfective projects

[4 marks]

b) What are the six (6) activities in a software project lifecycle and how does the object-oriented approach affect these software development activities?

[4 marks]

c) Describe the *functional*, *quality*, *platform*, and *process* requirement statement types used in software design analysis. Then choose one (1) requirement statement from the list below which matches each type in your answer and justify the reasons why you think that statement is that particular requirement type.
   1. The Student Information System must respond to inputs within 5 seconds.
   2. Output generated must formatted to print to a large scale plotter.
   3. All variables will be stored using String datatypes using all caps names.
   4. A library-sort algorithm must be used to sort submitted work by CPU workload.
   5. The program can only run in ARM and certain x86 devices.
   6. Delivery of a working system should be within 90 days from the agreed specifications

[3 marks × 4]

d) One of the five principles of proper object-oriented design is the "*Liskov Substitution Principle*". Explain using a proper example (with help of code or UML diagram if necessary) the consequences when applying AND when ignoring this principle in software design.

[5 marks]

Continued...

Question 3

a) Explain the differences between UML structural and behavioral diagrams and provide one (1) example of each.

[4 marks]

b) Using proper UML class diagram notation, show any four (4) relationships that can exist between two class objects in an object-oriented program.

[2 marks × 4]

c) Convert the following Java code into its equivalent UML class diagram.

```
public class Car {
    protected String model;
    Car(String mdl){model = mdl;}
    public void displayPrice() {
    }
    public String getModel() {
        return model;
    }
    public void setModel(String model) {
        this.model = model;
    }
}

public class MPV extends Car {
    private int seatCapacity;
    private double tankSize;
    MPV(String mdl, double tank, int seat){
            super(mdl);
            seatCapacity=seat;
            tankSize=tank;
    }
    public void drive() {
    }
    public void crash() {
    }
}
```

[7 marks]

d) A car with manual transmission usually has five *gears* for different speeds of forward travel plus one for reverse. A driver changes (or upshifts) gears in order when speed increases but can lower (or downshift) to any smaller gear when needed. The reverse is never applied when the forward gears are in use.

Using a nested state diagram, illustrate the state of the transmission gears in a car.

[6 marks]

Continued...

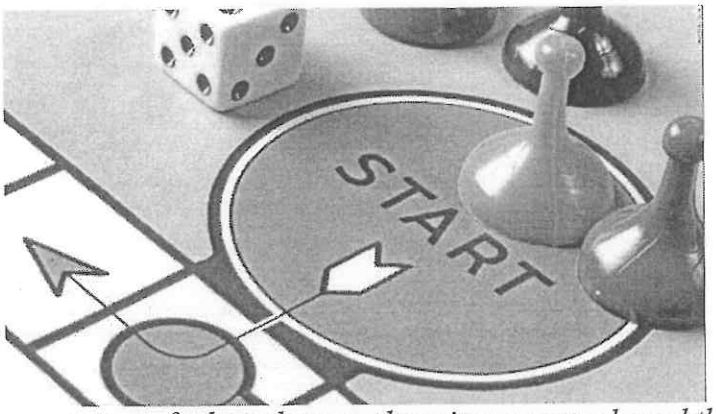
Question 4

a) Briefly describe the following design patterns in terms of the type of pattern and how the system objects will operate when implemented

i. Observer design pattern
ii. Builder design pattern
iii. Façade design pattern

[3 marks×3]

b)



"In an implementation of a board game, there is one game board that will keep track of many pieces of information. There should only be one game board. All the other objects in the system, when referring to a game board, should end up with the same game board"

What design pattern would you use for this scenario above? Briefly give an example of the design pattern chosen and also explain why it was chosen. You are not required to provide a class diagram, but you may use one in your explanation if you want to.

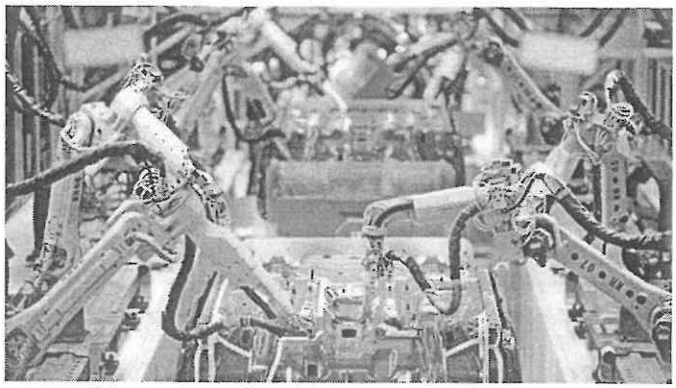
Warning: No marks will be given if you do not explain why you chose this design pattern. There could be more than one possible design pattern, and your explanation will partially determine if your choice is reasonable or not.

[8 marks]

c) In a manufacturing system, the system needs to, when started up, survey all the machines in the system and get their configurations. Then, different objects will be customized for different processes, and they will use these configurations. Depending on the processes, these objects may get further customized.

The survey process is relatively time-consuming (on the order of seconds) while the process-minding objects may get spawned off (reproduced) fairly regularly as new jobs appear in the manufacturing queue.

Continued...



What design pattern would you use for this scenario above? Briefly example the design pattern chosen and also explain why it was chosen. You are not required to provide a class diagram, but you may use one in your explanation if you want to.

Warning: No marks will be given if you do not explain why you chose this design pattern. There could be more than one possible design pattern, and your explanation will partially determine if your choice is reasonable or not.

[8 marks]

End of paper